ULTRIX

Guide to Disk Maintenance

Order Number: AA-ME93B-TE

June 1990

Product Version: ULTRIX Version 4.0 or higher

This guide describes how to use and maintain the disks on your system.

digital equipment corporation
maynard, massachusetts

Contents

3 Disk Partitioning

4 Bad Block Replacement

Figures

Tables

About This Manual

This manual provides you with information on using and maintaining disks. It presents background information on file systems and system disks, describes how to manage disk space and how to partition disks, and explains bad block disk error recovery techniques for Digital Standard Architecture (DSA) and Small Computer System Interconnect (SCSI) disks.

Audience

The *Guide to Disk Maintenance* is written for the person responsible for managing and maintaining an ULTRIX system. It assumes that this individual is familiar with ULTRIX commands, the system configuration, the system's controller/drive unit number assignments and naming conventions, and an editor such as `vi` or `ed`. You do not need to be a programmer to use this guide.

Organization

This manual consists of four chapters and an appendix.

Chapter 1: Disk Organization
Provides an overview of disks and file systems and describes default partition values for system disks.

Chapter 2: Managing Disk Space
Explains how to check disk space and disk use and how to obtain additional space on a disk.

Chapter 3: Disk Partitioning
Explains how to partition disks using the `chpt` command.

Chapter 4: Bad Block Replacement
Describes how to detect and replace bad blocks on DSA and SCSI disks and how to recover data from a forced error on DSA disks.

Appendix A:	Device Mnemonics Identifies and defines the mnemonics that are used to attach any hardware or software device to your system.

Related Documents

You should have the hardware documentation for your system and peripherals.

Conventions

The following conventions are used in this manual:

Kbyte	Throughout the text, the abbreviation Kbyte is used for kilobyte. One kilobyte equals 1024 bytes.
Mbyte	Throughout the text, the abbreviation Mbyte is used for megabyte. One megabyte equals 1,048,576 bytes.
rlogin	In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.
%	The default user prompt is your system name followed by a right angle bracket. In this manual, a percent sign (%) is used to represent this prompt.
#	A number sign is the default superuser prompt.
`user input`	This bold typeface is used in interactive examples to indicate typed user input.
`system output`	This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.
filename	In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents.
[ ]	In syntax descriptions and function definitions, brackets indicate items that are optional.
. . .	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
. . .	In syntax descriptions and function definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.

`cat`(1)	Cross-references to the *ULTRIX Reference Pages* include the appropriate section number in parentheses. For example, a reference to `cat`(1) indicates that you can find the material on the `cat` command in Section 1 of the reference pages.
RETURN	This symbol is used in examples to indicate that you must press the named key on the keyboard.
CTRL/*x*	This symbol is used in examples to indicate that you must hold down the CTRL key while pressing the key *x* that follows the slash. When you use this key combination, the system sometimes echoes the resulting character, using a circumflex (^) to represent the CTRL key (for example, ^C for CTRL/C). Sometimes the sequence is not echoed.

1 Disk Organization

This chapter provides background information about file systems, paging and swapping, and system disk partition defaults. ULTRIX supports various file systems, including the ULTRIX File System (UFS) and the Network File System (NFS). This chapter describes the UFS. For a discussion of NFS, see the *Guide to the Network File System.*

The following sections describe:

- Partitions
- Paging and swapping
- File systems
- System disk layout

1.1 Partitions

A disk consists of storage units called sectors, which are usually 512 bytes. A sector is addressed by the logical block number (LBN). The LBN is the basic unit of the disk's user-accessible data area that you can address. The first LBN is numbered 0, and the highest numbered LBN on a disk is numbered one less than the number of LBNs in the user-accessible area of the disk.

Sectors are grouped together to form up to eight partitions. However, disks differ in the number of partitions and in the size of partitions. For a listing of supported disks and partition sizes, refer to `hp`(4), `ra`(4), `rb`(4), `rd`(4), `rk`(4), `rl`(4), `rx`(4), and `rz`(4) in the *ULTRIX Reference Pages.* The `/etc/disktab` file contains a list of supported disks and partition sizes for the system. Refer to `disktab`(5) in the *ULTRIX Reference Pages* for more information.

Disk partitions are logical divisions of a disk that allow you to put files of the same type into separate areas of varying sizes. Partitions have default sizes that depend on the type of disk, but the partition sizes can be altered during installation or by using the `chpt` command. Partitions can be labeled `a` to `h`, but some disks have fewer than eight partitions. Also, the allocated space for a partition can overlap another partition. For example, partition `h` on an RA60 disk can overlap the `c`, `d`, `e`, and `f` partitions.

Note

> Although partitions can overlap, a properly partitioned disk should not use overlapping partitions.

Partitions can be used for system operations such as paging and swapping, and they can also be set up to hold user data in structures called file systems. File systems organize data stored on the disk, and they have a hierarchical structure of directories and files. Several file systems can reside on the same disk. A file system can be

smaller but not larger than the partition size.

Note

> A partition that is devoted to system operations must not overlap another partition.

By selecting the file systems to be placed in each partition, you can monitor the growth and activity of the disk. If you have multiple disks, you can divide users across the disks for better input and output response. You can also accommodate growth by placing the file systems you expect to expand on partitions that are large enough to accommodate that growth.

1.2 Paging and Swapping

The operating system needs disk space for paging and swapping to provide a time-sharing environment and virtual memory for users. Swapping occurs when there is not enough physical memory for one or more user programs (processes) in the system. Paging occurs when it is necessary to bring data into memory, to copy data out of memory, or to accommodate other processes that require physical memory.

The operating system provides a paging subsystem, so programs do not have to be written to fit within certain boundaries of physical memory. The hardware processes programs in lengths called pages. If a particular page of memory is not modified during processing, it is overwritten when that memory is needed for something else. When the page of memory is needed again, it can be read in from the original file on disk. If a page of memory is modified during processing, it is written out to disk so that the copy on disk reflects the page in memory. This writing and reading of modified pages of memory is one of the ways the system uses the paging and swapping area on disk.

The operating system's swapping subsystem keeps as many processes in physical memory as it can and lets each one execute during its own particular time slice. Swapping occurs when a new process needs to be loaded and executed, but there is no room for it in memory. An old process is swapped out to disk to wait its turn for processing again. When that time comes, the process is swapped back into memory to continue processing.

1.3 File Systems

This section describes how to access and manage your file systems. For information about backing up and restoring file systems, refer to `dump`(8) and `restore`(8) in the *ULTRIX Reference Pages* and to the *Guide to System Backup and Restore.*

A file system is described by its superblock, which contains the basic parameters of the file system, such as the number of data blocks, the maximum number of files, and a pointer to the list of free blocks. Partition `c` describes the entire disk. Partition information for the entire disk resides in the superblock of the `a` partition. The superblock is built when the file system is created with `newfs`, and it is located at a fixed offset from the beginning of the partition. The superblock is copied to other locations in the partition to create backup superblocks and to ensure that a copy exists if a disk error occurs.

Within a file system are files, such as data files and directory files. Each file is represented on the disk by an inode, which contains information describing the owner of the file, modification and access times, and indices pointing to the file's data

blocks.

Information about the file systems on your system is contained in the /etc/fstab file. The file is read at boot time as a reference for mounting the file systems, and each entry describes a known file system. The order of the entries is important because the mount command processes the entries in order. For example, a mount command will fail if you try to mount a file system on a directory that does not exist because the file system it resides on is unmounted.

1.3.1 Creating File Systems

For example, to create a file system on a disk partition, you should use the newfs command. The syntax of the newfs command is:

/etc/newfs [-v] [-n] [-N] [mkfs-options] *special disktype*

The *special* is the unmounted, raw device. For each disk partition, there is a corresponding raw device. For example, to create a file system on partition a of the device rd0, you specify /dev/rrd0a for the *special.*

For more information about the syntax of the newfs command, refer to newfs(8) in the *ULTRIX Reference Pages.*

Note

> When creating a file system, the newfs command destroys all existing data on the partition. You should back up important data before you use the newfs command.

Before creating a file system on a partition, you may want to determine the size of the partitions on the disk by using the chpt command with the -q option and the raw device name. For more information on the chpt command, refer to Chapter 3 and chpt(8) in the *ULTRIX Reference Pages.*

For example, to determine the partition layout on an RD32 disk, drive 0, use the following command:

```
# /etc/chpt -q rrd0a  RET
/dev/rrd0a
No partition table found in superblock...
using default table from device driver.
Current partition table:
partition     bottom        top       size     overlap
    a              0      15883      15884     c,f,h
    b          15884      31508      15625     c
    c              0      83235      83236     a,b,d,e,f,g,h
    d          31509      57371      25863     c,g
    e          57372      83235      25864     c,g
    f              0          0          0     a,c,h
    g          31509      83235      51727     c,d,e
    h              0          0          0     a,c,f
#
```

You should create a file system with the newfs command if one does not exist in the a partition of the device, because partition information for the disk resides in the superblock of the a partition.

For example, to create a file system on partition a on an RD32 disk, drive 0, use the following command:

```
# /etc/newfs -vn /dev/rrd0a rd32 [RET]
/etc/mkfs /dev/rrd0a 15884 17 6 8192 1024 16 10 60 2048
Warning: 40 sector(s) in last cylinder unallocated
/dev/rrd0a:     15872 sectors in 156 cylinders of 6 tracks, 17 sectors
        8.1Mb in 10 cyl groups (16 c/g, 0.84Mb/g, 256 i/g)
super-block backups (for fsck -b#) at:
 32, 1696, 3360, 5024, 6688
#
```

You should note the LBNs of the backup superblocks, so you can access a backup superblock if the superblock is corrupted.

To create a file system on partition `g` on an RD32 disk, drive 0, use the following command:

```
# /etc/newfs /dev/rrd0g rd32 [RET]
```

Before you mount the file system you created, use the `fsck` command to check the integrity of the file system. Refer to Section 1.3.3 and `fsck`(8) in the *ULTRIX Reference Pages* for more information about using the `fsck` command.

For example, to check a file system on partition `g` on an RD32 disk, drive 0, use the following command:

```
# /etc/fsck /dev/rrd0g [RET]
```

1.3.2 Mounting File Systems

File systems must be mounted in order to be accessed by the system or user. You use the `mount` command to mount a file system or a remote directory onto a local directory mount point. The mount point must already exist in order for a file system or directory to be mounted on it.

The `umount` command unmounts a file system or directory that was previously mounted on a directory. A general description of the `mount` and `umount` commands is provided in this manual. Additional descriptions of `mount` and `umount` syntax and options for the NFS and UFS file systems are provided in the *ULTRIX Reference Pages.*

The general formats of the `mount` and `umount` commands are as follows:

/etc/mount [-a] [-f] [-o] [-r] [-t] [-v] [*device*] [*directory*]

/etc/umount [-a] [-v] [*device*] [*directory*]

If invoked without options, `mount` prints the list of mounted file systems. Refer to `mount`(8) in the *ULTRIX Reference Pages* for information about options for the `mount` and `umount` commands.

Each invocation of the `mount` command announces to the system that a file system is present on the specified *device.* The file system may be local or remote. If the file system is remote, part of a file system tree can be mounted. The *directory* must exist as a directory file. It becomes the name of the newly mounted file system.

The `umount` command announces to the system that the file system previously mounted on the specified directory is to be removed. Only the person who mounted a particular file system or the superuser can unmount the file system.

An example of mounting and unmounting a file system is:

```
# mount RET                                    1
/dev/ra1a on / type ufs
/dev/ra1g on /usr type ufs
# ls /disk52a RET                              2
#
# mount /dev/ra0a /disk52a RET                 3
# mount RET                                    4
/dev/ra1a on / type ufs
/dev/ra1g on /usr type ufs
/dev/ra0a on /disk52a type ufs
# ls /disk52a RET                              5
bin         genvmunix     opr          ultrixboot    vmb.exe
dev         lib           pcs750.bin   upgrade       vmunix
doclib      lost+found    sys          usr
etc         mnt           tmp          var
# umount /dev/ra0a RET                         6
# mount RET                                    7
/dev/ra1a on / type ufs
/dev/ra1g on /usr type ufs
# ls /disk52a RET                              8
#
```

The previous example shows the following steps:

1. The `mount` command displays all the current mounted file systems.
2. The `ls` command displays the empty contents of the `/disk52a` directory because the file system is unmounted.
3. The `mount /dev/ra0a /disk52a` command mounts the local file system `/dev/ra0a` onto the local directory `/disk52a`.
4. The `mount` command displays all the current mounted file systems.
5. The `ls /disk52a` command displays the contents of the `/disk52a` directory. The directory is now an access point to the files contained in the file system `/dev/ra0a`.
6. The `umount /dev/ra0a` command unmounts the file system.
7. The `mount` command shows that `/dev/ra0a` file system is no longer mounted.
8. The `ls /disk52a` command shows that the directory is empty because it is no longer an access point to a mounted file system.

1.3.3 Checking File Systems

The `fsck` command checks and corrects file systems to help ensure a reliable environment for file storage on disks. File system inconsistencies such as unreferenced inodes, missing blocks in the free list, or incorrect counts in the superblock can be corrected by `fsck`. The system runs `fsck` noninteractively during the system boot, and it makes corrections that can be made safely. If it encounters an unexpected inconsistency, `fsck` exits, leaves the system in single-user mode, and displays a recommendation that you run `fsck` manually, which allows you to answer yes or no to the corrective actions that `fsck` displays.

File systems can become corrupted in many ways, such as improper shutdown procedures, hardware failures, and power outages and surges. A file system can also become corrupted if you physically write protect a mounted file system, take a mounted file system off line, or if you do not synchronize the system before you shut the system down.

The format of the fsck command is:

/etc/fsck [-b] [-y] [-n] [-p] [-P] [*filesystem* ...]

The fsck command used without the -p or -P option allows you to interactively repair inconsistent file systems; you are prompted before each correction is attempted.

Note

> A file system must be unmounted before you use fsck. However, you should not try to unmount the root file system on a running system. To check the root file system, you must be in single-user mode, and you must not use the raw device.

Refer to fsck(8) in the *ULTRIX Reference Pages* for information about options for the fsck command.

An example of using the fsck command is:

```
# mount RET                              1
/dev/ra0a on / type ufs
/dev/ra0d on /var type ufs
/dev/ra0e on /var/adm/crash type ufs
/dev/ra1a on /tmp type ufs
/dev/ra3a on /usr type ufs
/dev/ra3h on /usr/teach type ufs
/dev/ra17a on /usr/local type ufs
/dev/ra18g on /usr/staff/r1 type ufs
#
# umount /dev/ra3h  RET         2
# /etc/fsck /dev/rra3h  RET          3
** /dev/ra3h
** Last Mounted on /usr/teach
** Phase 1 -- Check Blocks and Sizes
** Phase 2 -- Check Pathnames
** Phase 3 -- Check Connectivity
** Phase 4 -- Check Reference Counts
** Phase 5 -- Check Cyl groups
4384 files, 37277 used, 89091 free (291 frags, 22200 blocks)
#
```

The previous example shows the following steps:

1. The mount command shows all the current mounted file systems.
2. The umount command unmounts the file system to be checked.
3. The fsck command checks the file system and displays diagnostic messages about the following:
    - Directory the file system was last mounted on
    - Consistency of the i-nodes
    - Directories that point to incorrect i-nodes
    - Errors regarding unreferenced directories
    - Link counts in directories and files
    - Bad, duplicate, or unreferenced free blocks and total free block counts
    - Information about the number of files and blocks in the file system

1.4 System Disk Layout

When you perform an advanced installation, the install procedure allows you to choose either the existing disk partition layout or the default system disk partition layout. The system disk is the disk that contains the `root` file system, and it may also contain other file systems and swap space. In addition, the system disk is the disk on which the installation software installs the ULTRIX subsets.

In general, the default system disk partitions are as follows:

- Partition `a` contains the `root` (/) file system.
- Partition `b` is reserved for use as a paging and swapping area and for crash dumps.
- Partition `g` is reserved for the `/usr` file system. The `var` area is a directory under the `/usr` file system. On RA90 disks, the default area for `/usr` is partition `d`.

The file system organization can simplify system management by placing volatile files, executable command files, and user files in separate directories. Separating files by content improves system debugging because files of the same type are located in the one area. Also, as you add software or as you add new users, you can keep track of disk space use and modify disk partitions when necessary.

1.4.1 Allocating Swap Space

The swap space is the area on a disk that is used by the operating system to transfer processes into and out of physical memory. The size of the swap space should be at least the size of your processor's main memory, and it should be two or three times the size of main memory if your system is going to run numerous processes at the same time. For example, a system that is primarily devoted to software development probably needs a swap space at least two times the size of main memory.

If the swap area is too small, the system cannot make use of all the available physical memory. If you receive messages indicating that there is not enough memory, you probably need a larger swap area.

Note

> You should avoid selecting partition `a` or any partition that overlaps partition `a` for use as a swap partition. Customized partition information for the disk resides in the superblock of the `a` partition. If a customized partition table has previously been defined for the disk, the information is destroyed when other data is swapped to partition `a`.

By default, the installation procedure automatically allocates partition `b` of the system disk for paging and swapping. If you determine that you need more paging and swapping space, you can use a second partition for paging and swapping. In this case, the kernel can use both partitions for its paging and swapping operations, and your system may run faster than it would with only one swap space.

You can allocate a second partition for swapping during an advanced installation or by making the necessary assignments in the configuration file. For information on how to define this configuration, refer to the *Guide to System Configuration File Maintenance*. In addition, you can also change the size of partition `b` on the system disk by using the `chpt` command as described in Chapter 3.

Local paging and swapping is available to clients who have a local disk and are operating in a diskless environment. Refer to the *Guide to Diskless Management Services* for details.

1.4.2 System Directories

The system needs certain directories for standard operations. This section describes these directories and their use.

The `root` (/) file system contains the following types of directories and files:

- The `/etc` directory contains maintenance programs, accounting files, important system programs executed from the shell level, and system initiation programs.
- The `/bin` directory contains utility programs executed from the shell level.
- The `/tmp` directory is used by various system and user programs, such as the `vi` editor, for temporary files.
- The /dev directory contains the `MAKEDEV` script and the system special files for devices, printers, and terminals.

The `/usr` file system contains the following types of directories and files that can be shared with client systems:

- The `/usr/adm` directory is for administrative information.
- The `/usr/lib` directory contains object library routines and documentation macros.
- The `/usr/ucb` directory contains utility programs.
- The `/usr/hosts` directory contains a file for each node on the local area network. The `/usr/hosts/MAKEHOSTS` command creates these files, which are symbolically linked to the remote shell program.
- The `/usr/man` directory contains the source for documents printed in the *ULTRIX Reference Pages.*
- The `/usr/users` directory is the default for the home directory for each of the system.

  In an environment where `/usr` is shared with client systems, `/usr/users` should be set up as a separate file system. If you installed your system with the advanced installation, you were provided with the option of creating a separate file system for `/usr/users.` For more information on the `/usr/users` file system, see the section User Directories.

The `/var` area contains volatile files, such as log files, that increase and decrease in size. When you perform the basic installation, files and directories that would normally reside in `/var` are located in `/usr/var,` which is symbolically linked to `/var.`

- The `/var/adm` directory contains data files generated by administrative programs such as system accounting and error logger. The data and files in `/var/adm` can vary widely across systems and over time. The `/var/adm/acct` file, for example, can easily grow by 5 Kbytes a day. In addition, `/var/adm` is the default location for the system crash directory, `/var/adm/crash.`

- The /var/spool directory contains files being held for spooling. For example, files for the line printer and mail messages are located in this directory.

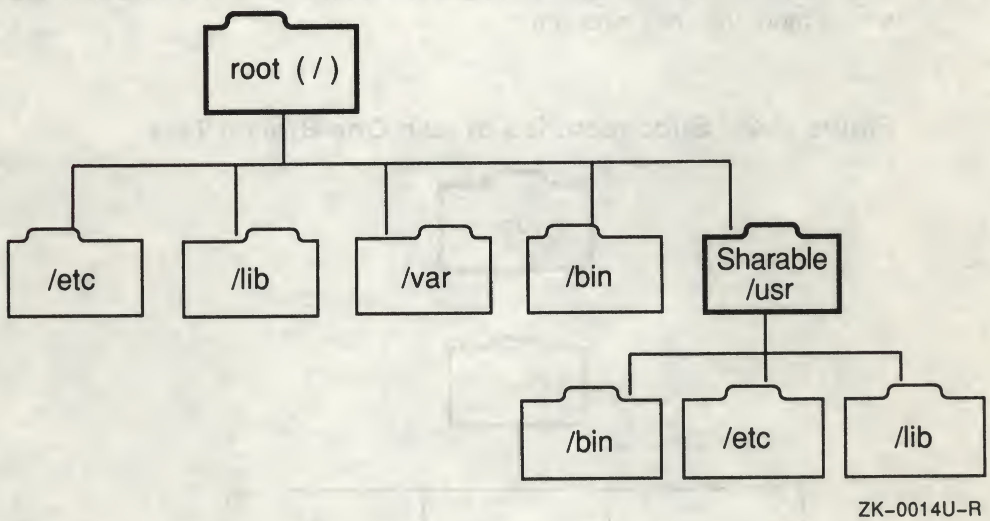
  You should monitor the variable growth of /var/spool to determine the amount of space that you need. If your users place heavy demands on these facilities, then allocate more space accordingly.

- The /var/adm/ris directory contains the remote installation environment.

If your processor is a server for diskless clients and you want the server and diskless clients to share the same /usr area, you should set up separate file systems for both /var and /usr/users. The files in the /usr file system are laid out to be read only, while files in the /var file system are volatile system-specific files such as /var/adm/shutdownlog. Diskless clients can share the information in the /usr file system with other diskless clients.

Figure 1-1 shows a file system organization that has a shareable /usr file system.

Figure 1-1: File System Organization



This file system organization provides the most secure access to shared files and simplifies system management tasks by limiting extremely dynamic files to one area. To set up separate file systems for /var and /usr/users, perform an advanced installation and follow the instructions in the *Advanced Installation Guide*.

1.4.3 User Directories

By default, the login (home) directory for users is subordinate to /usr/users. You can change this default so that /usr/users contains one or more subdirectories for your users.

In general, any user who logs in to the system has access to the files in their home directory.

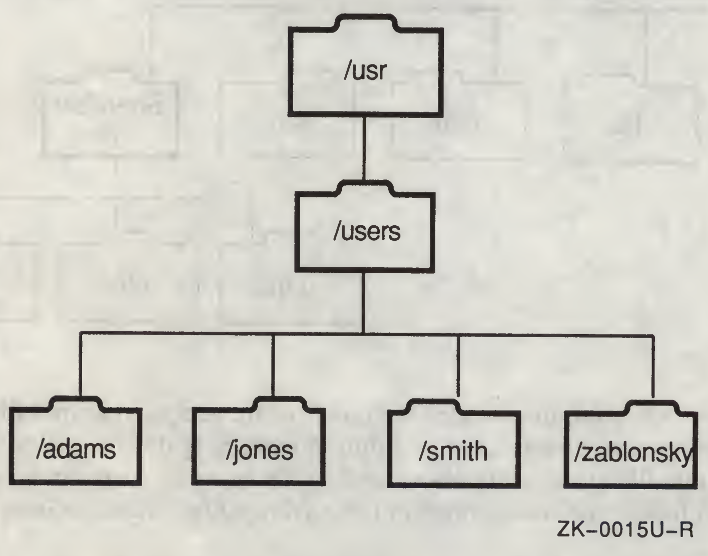
- On local systems, the default locations for individual home directories are in subdirectories of /usr/users. For example, a user named mjadams could

have a home directory with the pathname: `/usr/users/mjadams`.

- On a remote client system that shares the `/usr/users` file system, users can mount the home directory on any established mount point.
- On client systems that run Yellow Pages, the home directory mount point on the client should have the same pathname as the pathname for the home directory on the server system. For example, if the pathname of the home directory for user rjones is `/usr/users/rjones`, you must create a directory for `/usr/users/rjones` on your client system as the mount point. Because Yellow Pages systems share the password file, the home directory pathname, userid, and groupid must match on the client and the server systems. For more information on Yellow Pages, refer to the *Guide to the Yellow Pages Service.*

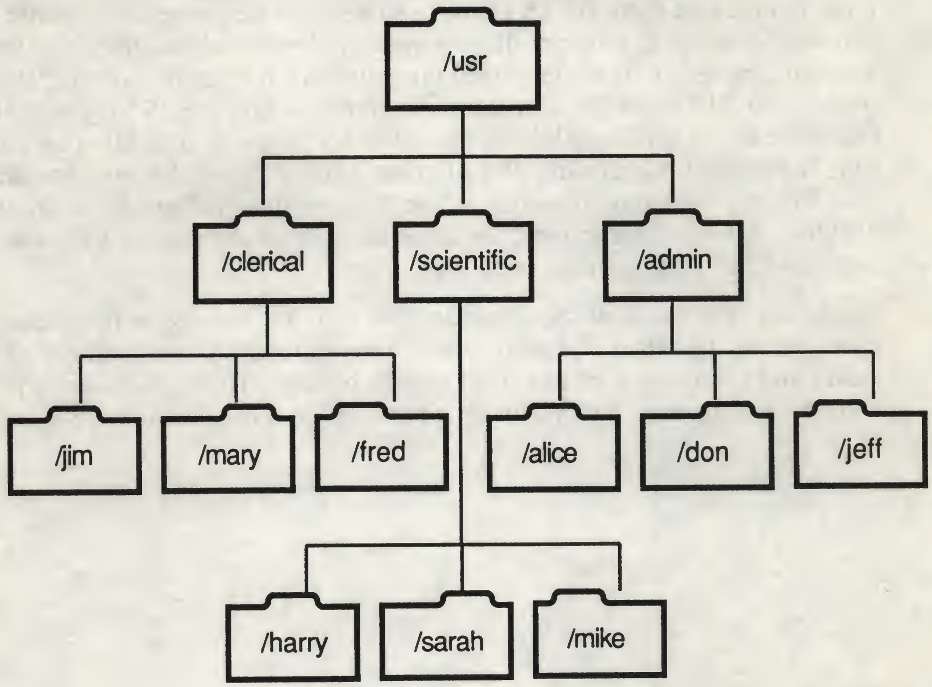
1.4.3.1 Changing the User Directory Tree Structure – If you want to change the default structure of the `/usr/users` directory, you must know how to set up directory tree structures, and you must understand the needs of your users. You can create a directory tree structure for users under the `/usr` directory with as few or as many branches as you want. You can, for example, create home directories for all users within one subdirectory, as shown in Figure 1-2. If the space required by all your users fits onto one disk partition, you do not have to divide the directories across more than one partition.

Figure 1-2: Subdirectories of /usr, One-Branch Tree



You may, however, require more complex allocations. You can organize user directories into subdirectories for different types of users, all with different requirements. This example is illustrated in Figure 1-3, where users are divided into three categories according to disk use: clerical, scientific, and administrative.

Figure 1-3: Subdirectories of /usr, Three-Branch Tree



If you need more space for a file system you can move it to another partition. You should use the `newfs` command to create new file systems on partitions that are large enough to hold them and any anticipated growth. For more information, refer to Chapter 2 and to `newfs`(8) in the *ULTRIX Reference Pages.*

You should note that the size of a file system is defined by the size of the partition on which it resides. If you put a small file system into a large partition, the file system would waste the excess disk space on that partition. Consequently, you must make sure that a large file system resides on a partition that is large enough to accommodate its size. For example, you might mount a large file system on partition `g` because this partition overlaps partitions `d`, `e`, and `f`. By using partition `g`, the file system has more room than if you use partition `d`, `e`, or `f`.

To review the current disk partition setup, use the `chpt` command with the `-q` option. For more information, see Chapter 3 and `chpt`(8) in the *ULTRIX Reference Pages.*

1.4.3.2 **Evaluating User Requirements –** If you have different types of users, such as clerical, scientific, and administrative, you may put each type onto a file system that can accomodate its growth and security needs. You should determine the requirements of the different users and then examine your resources before setting up the file systems.

For example, assume that there are 10 clerical workers on a system, and they each have at least 25 files on hand at any given time. These users generate `nroff` source and output files for memos, papers, and reports, which require an average of 30 Kbytes each.

The average load, in this case, is 7.5 Mbytes, which you could put on an 8-Mbyte partition. To allow for growth and periods of high production, you could increase the size of an 8-Mbyte partition or allocate the file system to a larger partition for this group.

Also assume that there are 15 graduate students in the group of scientific users and that they generate C program files — source, object, and executable. These users keep an average of 10 modest-sized programs (15 Kbytes for source, 10 Kbytes for object, and 20 Kbytes for executable programs), totaling 6.75 Mbytes. Also assume that there is one professor who keeps 10 of his programs and data bases at any given time (he generates huge data files of about 1,000 Kbytes); he would require around 11.2 Mbytes. The total requirement for the scientific staff would be around 18 Mbytes. Allowing for growth, the scientific staff might require a file system on a medium-sized partition on a large disk.

In addition, the needs of the administrative staff are as large as the scientific users' needs (about 18 Mbytes), but they have a special security requirement. They keep salary and personnel files that must remain private. To maintain this privacy, they should be assigned a file system on a partition on a disk with removable media.

Managing Disk Space 2

This chapter discusses methods of monitoring file system use and managing file system data. The first section in this chapter explains how to check your disk space and disk use. The second section describes different methods you can use to obtain additional disk space.

2.1 Monitoring File System Use

To ensure an adequate amount of free disk space, regularly monitor the disk use of your configured file systems.

- Check available free space by using the `df` command.
- Check disk use by using the `du` command with the `-s` option.
- Verify disk quotas (if imposed) by using the `quota` command.

You should become the superuser to perform these tasks. You can issue some commands, such as the `quota` command, only if you are the superuser.

2.1.1 Checking Available Free Space

To ensure sufficient free space for your configured file systems, you should regularly check the amount of free disk space in all of the mounted file systems. To see how much free disk space is available, use the `df` command. The syntax of the `df` command is:

df [-i] [-n] [*filesystem* ...] [*file* ...]

Without any arguments or options, `df` displays both the used disk space and the free disk space on all of the file systems that are mounted. For each file system, the `df` command reports the file system's configured size in Kbytes, which are 1024 bytes. The command displays the amount presently used, the amount presently available (free), the percentage used, and the directory on which the file system is mounted.

The `-i` option reports the number of used and free inodes. The number of inodes controls the number of files that can exist in a file system.

The `-n` option does not update the file system statistics stored in memory. Instead, it returns whatever statistics are stored in memory. This prevents `df` from hanging in the event that a server containing the specified file system is down.

If you specify *file system,* the amount of available disk space on the file system is displayed.

If you specify *file,* the amount of available disk space on the file system that contains the file is displayed.

If a device is given that has no file systems mounted on it, `df` displays the information for the root file system.

Note

You cannot use the df command to find free space on an unmounted file system using the block or character special device name. Instead, use the dumpfs command.

The following example displays disk space information about all the mounted file systems:

```
# df [RET]
Filesystem     total    kbytes  kbytes  percent
   node        kbytes    used    free   used    Mounted on
/dev/ra0a       15551    9030    4966    65%    /
/dev/ra2e       48196    1792   41585     4%    /tmp
/dev/ra0h      300558  137618  132885    51%    /usr
/dev/ra8g       62421   28027   28152    50%    /var
/dev/ra7g      105631   20244   74824    21%    /var/spool
/dev/ra6d      571444  341801  172499    66%    /usr/uws3
/dev/ra6e      571444  428207   86093    83%    /usr/uws4
/dev/ra1d       29615     106   26548     0%    /var/tmp
/dev/ra1e      151099   88897   47093    65%    /var/adm
/dev/ra6f      404131  115270  248448    32%    /usr/staff
```

The following example displays disk space information about the file system /dev/ra6f:

```
# df /dev/ra6f [RET]
Filesystem     total    kbytes  kbytes  percent
   node        kbytes    used    free   used    Mounted on
/dev/ra6f      404131  115270  248448    32%    /usr/staff
#
```

The following example displays disk space information about the file system where /usr/staff/maryb is located:

```
# df /usr/staff/maryb [RET]
Filesystem     total    kbytes  kbytes  percent
   node        kbytes    used    free   used    Mounted on
/dev/ra6f      404131  115270  248448    32%    /usr/staff
#
```

File systems are usually configured with a minimum percentage of free space already established. When constructing a new file system, the newfs command reserves a minimum percentage of the space. The default is 10 percent. This default percentage allows for a report in excess of 100 percent. When interpreting the free space report, therefore, you should look for significant changes in file system disk use.

For more information on the df command and its options, refer to df(1) in the *ULTRIX Reference Pages.*

2.1.2 Checking Disk Use

If you determine that a file system has insufficient space available, check how its space is being used. The du command pinpoints disk space allocation by directory. With this information you can decide who is using the most space and who should free up disk space. The syntax of the du command is:

du [-a] [-s] [*name...*]

The du command gives the number of Kbytes contained in all files and, recursively, directories contained in each specified file *name*. If *name* is not specified, the current working directory is used.

The -a option displays the disk usage for each file.

The -s option displays a summary total only.

If you do not specify either the -a or the -s option, an entry is generated for each directory only.

The information about blocks used is reported in the same unit as the file system's fragment size, which is established when the file system is created. The default fragment size is 1024 bytes.

The following example displays a summary of blocks used by all main subdirectories in the /usr/users directory:

```
# du -s /usr/users/*  RET
440      /usr/users/barnam
43       /usr/users/broland
747      /usr/users/frome
6804     /usr/users/morse
11183    /usr/users/rubin
2274     /usr/users/somer
#
```

From this information, you can determine that user rubin is using the most disk space.

The following example displays the space used by each directory and subdirectory:

```
# du /usr/users/rubin  RET
104  /usr/users/rubin/V4.0/info
2713 /usr/users/rubin/V4.0
2    /usr/users/rubin/online/inof
251  /usr/users/rubin/online
   .
   .
   .
1    /usr/users/rubin/Mail/drafts
1    /usr/users/rubin/Mail/wastebasket
4    /usr/users/rubin/Mail
11183/usr/users/rubin
```

The following example displays the space used by each file and subdirectory in the directory /usr/users/rubin/online:

```
# du -a /usr/users/rubin/online  RET
1    /usr/users/rubin/online/inof/license
2    /usr/users/rubin/online/inof
7    /usr/users/rubin/online/TOC_ft1
16   /usr/users/rubin/online/build
   .
   .
   .
3    /usr/users/rubin/online/AQdocinfo.boil
40   /usr/users/rubin/online/build.ps
4    /usr/users/rubin/online/AQft2.boil.nroff
251  /usr/users/rubin/online
```

You can also use the quot command to list the number of blocks in the named file system currently owned by each user.

The syntax of the quot command is:

/etc/quot [-n] [-c] [-f] [*filesystem*]

The -n option lists all the files and their owners.

The -c option prints three columns giving the file size in blocks, number of files of that size, and cumulative total of blocks in that size or smaller file.

The -f option prints the amount of space and a count of the number of files owned by each user.

If no *filesystem* is specified, a default name is used.

The following example displays the number of blocks used by each user and the number of files owned by each user in the /dev/ra0h file system:

```
# /etc/quot -f /dev/ra0h  RET
/dev/ra0h:
50071    2374   cjp
37630    1905   walters
37530   14584   sullivan
29793    1413   hobard
28394    1282   bfc
25482    5234   mallet
25369    2811   jean
17636    1255   wu
16074    2568   kelly
 8542    1434   apple
 1084     139   uwslib
  775      20   andrew
   12       7   root
    4       4   doyle
    3       2   #1038
#
```

For more information on checking disk space use, refer to du(1) and quot(8) in the *ULTRIX Reference Pages*.

2.1.3 File System Quotas

Quotas can be applied to file systems to establish a limit on the number of blocks and inodes (or files) that a user can allocate. A separate quota can be set for each user on each file system. You may want to set quotas on file systems that contain home directories, such as /usr/users, because the sizes of these file systems can increase more significantly than other file systems. You should avoid setting quotas on /tmp.

You can set a soft limit and a hard limit, which cannot be exceeded. When you exceed a soft limit, a warning is displayed. If you reach a soft limit, you are warned every time you log in. A soft limit eventually becomes a hard limit if you log into the system three consecutive times and if the soft limit is exceeded each time you log in.

If you exceed either the final soft limit warning or the hard limit warning, you must regain enough space to be below the quota limits by either removing files or moving files to another file system. To recover from over-quota conditions, you can abort the activity in progress on the file system, remove enough files to be below the quota limit, and restart the activity.

However, if you are in an editor, you do not want to abort the editor or write the file because data may be lost. Instead, you can use the editor exclamation point (!) shell escape command to remove files. You can also write the file to another file system such as `/tmp`, remove files from the file system whose quota you exceeded, and then move the file back to that file system.

To activate quotas on a file system, perform the following steps:

1. Configure the system to include the disk quota sub-system by editing the system configuration file `/usr/sys/conf/`*hostname* to include the following line:

   ```
   options        QUOTA
   ```

2. Use the `quotacheck` command with the `-f` option to create a quota file where the quota subsystem stores current allocations and quota limits. The syntax is:

   quotacheck -f *filesystem*

   The *filesystem* is the file system on which you are activating quotas.

3. Edit the `/etc/fstab` file and change the third field of the file system's entry to `rq` (read and write with quotas imposed). An example of a file system's `/etc/fstab` entry with quotas imposed is:

   ```
   /dev/ra4g:/usr/users/r3:rq:1:4:ufs::
   ```

4. Use the `edquota` command to activate the quota editor and create a quota entry for each user. The syntax of the `edquota` command is:

   edquota [-p *proto-user*] *user...*

   For each user you specify, `edquota` creates a temporary ASCII file that you edit by using the `vi` editor. Edit the file to include entries for each file system with quotas enforced and the soft and hard limits for blocks and inodes (or files). The syntax of an entry in the `edquota` temporary file is:

   fs *filesystem* blocks (soft = #, hard = #) inodes (soft = #, hard = #)

   The *filesystem* is the file system that has quotas imposed. The # is the soft and hard limit for the number of blocks and inodes (or files).

   If you specify more than one user in the command line, the edits will effect each user. If you specify the `-p` *proto-user* option and one or more users, the `edquota` will use the quotas of the specified *proto-user* for the quotas of the user or users that you specify. This allows you to quickly set up quotas for groups of users.

   An example of using the `edquota` command is:

   ```
   # edquota kms  RET
   fs / blocks (soft=0, hard=0) inodes (soft=0, hard=0)
   fs /usr blocks (soft=0, hard=0) inodes (soft=0, hard=0)
   fs /usr/users blocks (soft=2000, hard=2500) inodes (soft=30, hard=50)
   fs /usr/class blocks (soft=1000, hard=1500) inodes (soft=20, hard=25)
   fs /usr/adm blocks (soft=0, hard=0) inodes (soft=0, hard=0)
   ~
   ~
   ~
   "/tmp/EdP.aa4019 5 lines, 461 characters
   #
   ```

5. Reboot to reconfigure the system.
6. Use the `quotaon` command to activate the quota system.

   You can include the `quotaon -a` command in the `/etc/rc.local` file. Refer to `quotaon(8)` in the *ULTRIX Reference Pages* for more information.

   If you want to turn quotas off, use the `quotaoff` command. Also, the `umount` command turns off quotas on a file system before it unmounts it. Refer to `quotaoff(8)` in the *ULTRIX Reference Pages* for more information.

2.1.4 Verifying Disk Quotas

If you are enforcing user disk quotas, you should verify your quota system periodically. You can use the following commands to compare the established limits with actual use: `quotacheck`, `quota`, and `repquota`.

The `quotacheck` command verifies that the actual block use is consistent with established limits. The `quotacheck` command should be run twice: when quotas are first enabled on a file system and after each reboot. The command gives more accurate information when there is no activity on the system.

The `quota` command displays the actual block use for each user in a file system. The command can only be executed by the superuser.

The `repquota` command displays both the actual disk use and quotas for the specified file system. For each user, the current number of files and the amount of space (in Kbytes) is displayed along with any quotas.

If you find it necessary to change the established quotas, use the `edquota` command, which allows you to set or change the limits for each user.

For further information on disk quotas, refer to `quotacheck(8)`, `quota(8)`, and `repquota(8)` in the *ULTRIX Reference Pages*, as well as the *ULTRIX-32 Supplementary Documents.*

2.2 Obtaining Additional Disk Space

When you have checked disk space and use, you may need to obtain additional space on the disk or on a file system. You can use the following methods, depending on your system and the needs of its users:

- Delete, archive, or compress files
- Move a file system to a larger partition
- Move files to another file system

Another solution is to the change a file system's partition table, as described in Chapter 3.

2.2.1 Deleting, Archiving, and Compressing Files

If you determine that a file system or disk needs more space, you should request that users remove any unused or obsolete files. Use the `du` command to determine the users with the most disk space. If you are using disk quotas, use the `repquota` command to determine the users who have reached or who have almost reached their quotas.

If there is still an insufficient amount of free space on a file system, request that users of that file system archive their infrequently used files. For further information on backing up and restoring files, see the *Guide to System Backup and Restore*.

You can also decrease the size of a file by using the `compress` command, which allows you to reduce a file's size by up to 50%. PostScript files, especially, can be easily compressed. However, you must use the `uncompress` command to utilize the file; because of this, compressing files works best on files that are only accessed periodically.

Refer to `compress(1)` in the *ULTRIX Reference Pages* for more information on compressing files.

2.2.2 Moving a File System

If you find that you are running out of space in a file system or that you have excess space in a file system, you may want to move the file system to another partition either on the same or on a different disk.

Note

It is not recommended that you move any of the file systems that contain directories such as `/usr` and `/var` or that contain the swap space.

To move a file system from one partition to another, follow these steps:

1. Determine the amount of space available on your file systems by using the `df` command.

   The following `df` command shows that the `/dev/ra69h` file system is using only 36% of its allotted space of 238015 Kbytes, and it shows that the `/dev/ra1e` file system is using nearly all of its allotted space of 180652 Kbytes:

```
# df  RET
Filesystem      Total     kbytes    kbytes    %
node            kbytes    used      free      used   Mounted on
/dev/ra0a        15551      9030      4966     65%    /
/dev/ra0h       300558    137618    132885     51%    /usr
/dev/ra1a        62421        36     56143      0%    /usr/projects
   .
   .
   .
/dev/ra69h      238015     77255    136959     36%    /usr/SRC
/dev/ra1d       180652    156255      6332     96%    /usr/staff/ab6
/dev/ra1e       180652    162365       222    100%    /usr/staff/ab7
#
```

   It appears that to better utilize the disk space, the `/usr/SRC` directory can be moved to (or mounted on) the `/dev/ra1e` file system, and the `/usr/staff/ab7` directory can be moved to (or mounted on) the `/dev/ra69h` file system to accomodate space needs and future growth.

2. Unmount the file systems that you want to move, check the integrity of the file systems with the `fsck` command, and perform a level 0 dump of the file systems with the `dump` command. For more information about the `dump` command, refer to `dump(8)` in the *ULTRIX Reference Pages* and to the *Guide to System Backup and Restore*. For more information about `mount` and `fsck`, refer to Chapter 1.

For example, to back up the file system `/dev/ra1e` to the default tape device `/dev/rmt0h`, use the following commands:

```
# umount /dev/ra1e [RET]
# /etc/fsck /dev/rra1e [RET]
# dump 0u /dev/ra1e [RET]
```

After you dump the file systems, you should check the file systems again with the `fsck` command.

For example, to check the `/dev/ra1e` file system, use the following command:

```
# /etc/fsck /dev/rra1e [RET]
```

You should check all the file systems that are affected.

3. If you are moving a file system to a device that does not have a file system in partition `a`, create a file system with the `newfs` command. Also, you must create a file system on the partition that you want to use if one does not exist. Refer to Chapter 1 and `newfs`(8) in the *ULTRIX Reference Pages* for more information about using the `newfs` command.

4. At this point, you should make sure that you have the following:

   - Backups of the data in all the file system that are affected
   - Unmounted file systems that have been checked with the `fsck` command

5. Mount the directory on a file system that can accomodate its size.

   For example, to mount the `/usr/SRC` directory on the file system `/dev/ra1e`, use the following command:

```
# /etc/mount /dev/ra1e /usr/SRC [RET]
```

6. Use the `cd` command to change your current working directory to the directory on the target file system. For example:

```
# cd /usr/SRC [RET]
```

7. Restore the file system to the new partition with the `restore` command.

   For example, to restore files from the default tape device to the current directory, use the following command:

```
# /etc/restore r [RET]
```

8. Remove the restore table file `restoresymtable` from the current directory, and then change your default directory to `root`:

```
# rm restoresymtable [RET]
# cd / [RET]
```

9. Unmount the restored file system, check the file system with the `fsck` command, and mount the file system. The file system will then be ready to use.

   For example, to unmount, check, and mount the `/dev/ra1a` file system, use the following commands:

```
# /etc/umount /dev/ra1e [RET]
# /etc/fsck /dev/rra1e [RET]
# /etc/mount /dev/ra1e /usr/SRC [RET]
```

10. Edit the /etc/fstab file to show the new mount information.

    For example, the new /etc/fstab entry for the /dev/ra1e file system is as follows:

    ```
    /dev/ra1e:/usr/SRC:rw:1:1:ufs::
    ```

2.2.3 Moving Files to Another File System

You can move files to another file system by using either of two methods. You can move all the files, or you can move individual files. However, before you move any files, make a level 0 dump of the entire file system.

To transfer all the files to a larger file system, restore the level 0 dump to the larger target file system.

To move individual files from one file system into another, use the tar command to archive the files you want to move to the default output media, and then restore them to the target file system.

For example, to move the file named file1 from the current working directory to the default tape device /dev/rmt0h, type:

```
# tar c file1 RET
```

In this example, the tar command creates file1 on the default output media.

After creating the files you want to move on the default output media, use the tar command with the -x and -p options to restore (extract) the files from the tape device to the current working directory on the target file system.

For example,

```
# tar xp file1  RET
```

This command extracts file1 from the default tape device to the current working directory using the pathname on the tar archive media. The -p option is a superuser option that causes tar to use the original protection code assigned to file1.

Disk Partitioning 3

This chapter provides the information you need to change the partition sizes of your disks. In general, you allocate disk space during the initial installation or when adding disks to your configuration. Usually, you do not have to alter partitions; however, there are cases when it is necessary to change the partition sizes on your disks to accommodate changes and to improve system performance.

The following sections describe:

- Preparing to change disk partitions
- Disk partition table
- Reviewing the current file system assignments
- Reviewing the current disk partition setup
- Changing partition sizes

For a review of the concepts pertaining to ULTRIX disks, see Chapter 1.

3.1 Preparing to Change Disk Partitions

Before making any changes to the partitions on your disks, you should perform the following steps:

1. Assess the number of file systems you need and the amount of space each requires, including potential growth space. Here you might ask:
   - Are users, files, and directories logically and optimally grouped within file systems?
   - Does this setup satisfy our current needs?
   - Does this setup satisfy our projected needs?
2. Analyze the statistics available to you regarding current system performance. Here you might ask:
   - Does the current disk partitioning setup optimize the average and peak demands of system users?
   - Would a different setup improve system performance?
3. Review the current disk partition setup and file system allocation. You need to know the following:
   - The device type and partition defaults for all of the disks
   - The current size of each partition on the disks
   - The current location of each file system on the disks

4. Back up all the file systems that exist on a disk that you want to reconfigure.

3.2 Disk Partition Table

The first time a disk is opened, the device driver copies the disk's default table in the driver into an internal table that the driver can use. The device driver then checks the superblock of the `a` partition of the disk to see if the disk has an existing partition table. If there is no partition table, the driver uses the default values.

If the device driver finds a partition table in the superblock, the driver copies that table into the disk's active partition table in the device driver. This overwrites the default values that were originally copied into the active table.

The active set of partition tables in the device driver are the ones that the system uses for each disk, whether a partition table exists in the disk superblock or not. Remember, the active set of partition tables in the device driver contains one table for each disk on the system for that driver. Each table is a copy of either the default table in the driver or the individual partition table in the superblock of the `a` partition of the disk.

There are two reasons why a disk might not have its own partition table:

- The `a` or `c` partition does not have a file system for a partition table to reside in.
- An individual partition table was never created for the disk.

3.2.1 Partition Table Values

The ULTRIX partitioning scheme allows you to use the values for the default partition tables or to create or modify one or all of the partition table values for individual disks. You can continue to use the default values in the active partition tables in the device driver, or you can modify the partition table for a particular disk. There are two ways to override the default values:

- Use the `chpt` command. For more information on this command, refer to `chpt`(8) in the *ULTRIX Reference Pages*.
- Mount a disk that already contains a different partition table in the superblock of its `a` partition.

For example, making a new partition table allows you to have two similar disks on the system with different partitions, one using the default values for that disk type, the other using its own partition table, which is set up according to your space needs.

3.2.2 Rules for Changing Partition Tables

Rules and guidelines to follow when changing disk partitions are as follows:

- You must have superuser privileges to use the `chpt` command.
- Back up all the file systems before changing the tables if there is any data on the disk. Changing a partition overwrites the old file system, destroying the data.
- You cannot change the offset, which is the beginning sector, or shrink any partition on a mounted file system or on a file system that has an open file descriptor.

- If you need only one partition on the entire disk, use the existing partition `c`.
- A disk's partition table always resides in the superblock of partition `a`. Therefore, you must have an `a` partition with a file system on it before you can change the tables.

  If you plan to change a partition other than `a`, and there is no file system in partition `a`, create a file system on partition `a` before changing any others. Use `newfs` to create a file system. Partition `a` must begin at the start of the disk (sector 0). Refer to Chapter 1 for information on using the `newfs` command.

  Note

  Be careful when you change partition tables; you could overwrite data on the file systems or make the system inefficient. If the partition tables become corrupted while you are changing the partition sizes, you can return to the default partition table by using the `chpt` command with the `-d` option.

3.3 Reviewing the Current File System Assignments

Before you change the partition tables or reallocate disk space, review the current file system assignments on your disk. To do this, use the `df` command.

The `df` command displays the current file system assignments, showing the partitions and space being used by the system. In addition, it displays the free space available on all mounted file systems listed in `/etc/fstab` and lists the partitions for each disk.

For example, output from a `df` command is:

```
# df RET
Filesystem   Total    kbytes   kbytes     %
node         kbytes   used     free      used    Mounted on
/dev/ra0a     15551    12555     1441     90%      /
/dev/ra0d     62421    29317    26862     52%      /usr
/dev/ra6a     15551     9030     4966     65%      /usr/projects
/dev/ra6d    103963    85657     7910     92%      /usr/staff
```

The `df` command gives you a broad view of the disk partition assignments that have been made, and it displays disk space use in Kbytes. Note that a disk without file systems mounted on it does not show in the `df` display.

For more information, refer to Chapter 2 and to `df`(1) in the *ULTRIX Reference Pages.*

3.4 Reviewing the Current Disk Partition Setup

Before changing the size of a disk partition, review the current assignment. To do this, use the `su` command to become the superuser and issue the `chpt` command as follows:

/etc/chpt –q *device*

The `chpt` command with the `-q` option allows you to view the partition sizes. The *device* variable specifies the disk you are checking.

You must specify the device with its directory name (`/dev`), followed by the raw device name, drive number, and partition `a` or `c`. For example, to check current

partition sizes on an RA82 disk, drive 0, type:

```
# /etc/chpt -q /dev/rra0a  RET
/dev/rra0a
Current partition table:
partition    bottom       top       size    overlap
    a             0     32767      32768    c
    b         32768     99457      66690    c
    c             0   1216664    1216665    a,b,d,e,f,g,h
    d         99458    319553     220096    c,g
    e        319554    539288     219735    c,g
    f        539289    977048     437760    c,g
    g         99458    977048     877591    c,d,e,f
    h        977049   1216664     239616    c
```

The bottom, top, and size of the partitions are in 512 byte sectors.

3.5 Changing Partition Sizes

The `chpt` command allows you to change the partition table of an individual disk without rebuilding the kernel and rebooting the system.

The format of the `chpt` command is:

/etc/chpt [-a] [-d] [-q] [-v] [[-p*x offset size*] . . .] *device*

For information on the `chpt` command, refer to `chpt(8)` in the *ULTRIX Reference Pages*.

This section explains the procedure for changing partition sizes and gives an example for changing partition sizes on an RA82 disk. The example shows how to change a partition table on a running system.

Note

> The `chpt` command does not allow you to change the partitions on an active (mounted) file system. Also, the command does not allow you to change the starting sector (offset) of the `a` partition.

The following example assumes that you have an RA82 disk on drive 6 of your system and that you want to customize it to your system's needs by expanding partition `d` to include all the space in partition `e`. Perform the following steps to expand partition `d`:

1. Use the `su` command to become superuser.
2. Use the `df` command to display disk space information about all the mounted file systems:

```
# df  RET
Filesystem   Total    kbytes   kbytes    %
node         kbytes   used     free      used    Mounted on
/dev/ra0a     15551    12555     1441     90%      /
/dev/ra0d     62421    29317    26862     52%      /usr
/dev/ra6a     15551     9030     4966     65%      /usr/projects
/dev/ra6d    103963    85657     7910     92%      /usr/staff
```

3. Display the current partition table.

   Use the `chpt` command with the `-q` option to see the status of the partition table. For example, to display the partition table for an RA82 disk, type:

```
# /etc/chpt -q /dev/rra6a RET
/dev/rra6a
Current partition table:
partition      bottom       top        size     overlap
    a              0      32767       32768     c
    b          32768      99457       66690     c
    c              0    1216664     1216665     a,b,d,e,f,g,h
    d          99458     319553      220096     c,g
    e         319554     539288      219735     c,g
    f         539289     977048      437760     c,g
    g          99458     977048      877591     c,d,e,f
    h         977049    1216664      239616     c
```

4. Unmount the active file systems.

   Use the `mount` command to see which file systems are mounted on which devices. For example:

   ```
   # /etc/mount RET
   ```

   The `mount` program responds with information such as the following:

   ```
   /dev/ra0a  on  /  type ufs
   /dev/ra0d  on  /usr  type ufs
   /dev/ra6a  on  /usr/projects  type ufs
   /dev/ra6d  on  /usr/staff  type ufs
   ```

   From this response you learn that the `/usr/staff` file system is mounted on partition `d` of the RA82 disk. This is the disk that you want to change. Unmount the file system. For example:

   ```
   # /etc/umount /dev/ra6d RET
   ```

5. Back up the disk.

   You should back up the entire disk before changing partition sizes. Use the `dump` command to back up the files to another media. For information on backing up a disk, refer to `dump`(8) in the *ULTRIX Reference Pages*.

6. Calculate the new partition parameters.

   This step shows how to expand partition `d` of an RA81 disk so that it encompasses partition `e`. By looking at the partition table generated by the `chpt` command, you can see that you need to extend the size of partition `d` to the ending sector of partition `e`. Therefore, the new size of partition `d` is found by using the following formula:

   ```
   (size of d) + (size of e) = (new size of d)
   ```

   Here is the calculation:

   ```
   220096 + 219735 = 439831
   ```

   The offset of partition `d` remains the same.

7. Change the partition parameters.

   This step shows how to use the `chpt` command with the `-p` option.

   The following example extends the length of partition `d` by the size of partition `e`:

```
# /etc/chpt -v -pd 99458 439831 /dev/rra6a  [RET]
/dev/rra6a
New partition table:
partition     bottom        top       size    overlap
    a              0      32767      32768    c
    b          32768      99457      66690    c
    c              0    1216664    1216665    a,b,d,e,f,g,h
    d          99458     539288     439831    c,e,g
    e         319554     539288     219735    c,d,g
    f         539289     977048     437760    c,g
    g          99458     977048     877591    c,d,e,f
    h         977049    1216664     239616    c
#
```

8. Make and mount file system.

   Construct a file system on the partition, using the `newfs` command. For example:

```
# /etc/newfs /dev/rra6d ra81  [RET]
Warning: 346 sector(s) in last cylinder unallocated
/dev/rra6d:245984 sectors in 345 cylinders of 14 tracks, 51 sectors
    125.9Mb in 22 cyl groups (16 c/g, 5.85Mb/g, 2048 i/g)
super-block backups (for fsck -b#) at:
 32, 11520, 23008, 34496, 45984, 57472, 68960, 80448, 91936, 103424,
 114912, 126400, 137888, 149376, 160864, 172352, 182816, 194304,
 205792, 217280, 228768, 240256,
Warning: partition table overriding /etc/disktab
```

   If you are ready to use the file system, remount it on the partition with the `mount` command. For example:

```
# /etc/mount /dev/ra6d /usr/staff  [RET]
```

9. Restore backed up files.

   Use the `restore` command. For information on the `restore` command, refer to `restore`(8) in the *ULTRIX Reference Pages.*

4 Bad Block Replacement

This chapter describes how the system detects disk errors and how you can manually replace bad blocks on disks.

The following sections discuss these topics:

- Automatic bad block replacement – how the ULTRIX operating system does dynamic bad block replacement
- Disk error messages – how the system detects, records, and reports disk errors and bad blocks on Digital Storage Architecture (DSA) and Small Computer System Interconnect (SCSI) disks
- The `radisk` and `rzdisk` utilities – how to manually replace, clear, and scan bad blocks on a disk

4.1 Automatic Bad Block Recovery

There are three strategies the system uses to recover from bad blocks: controller-initiated, host-initiated, and media replacement. The controller-initiated strategy applies only to DSA disks. The controller-initiated and host-initiated strategies involve replacing a bad block with a good block reserved for this purpose by the system. The third strategy involves replacing the medium. Device support for these three strategies is summarized in Table 4-1.

Table 4-1: Device Support for Bad Block Strategies

Strategy	Controllers	Devices
Controller-initiated	RQDX1, RQDX2, RQDX3, KFQSA HSC50, HSC70, HDM70	RD31, RD32, RD51, RD52, RD53, RD54, RF30, RF31, RF71
	Small VAX disk controller*	RD31, RD32, RD53, RD54
Host-initiated	UDA50A, KDA50, KDB50	RA60, RA80, RA81, RA82 RA70, RA90, RA92
	SCSI	RZ22, RZ23, RZ24 RZ55, RZ56, RZ57
Media replacement	RQDX1, RQDX2, RQDX3	RX50, RX33
	RUX50	RX50
	SCSI	RX23
	Small VAX disk controller*	RX50, RX33, RX23

*The small VAX disk controller is used by the VAXstation 2000 and MicroVAX 2000 processors. The software driver for the small VAX disk controller emulates RQDX3-style controller-initiated bad block replacement.

The controller-initiated and the host-initiated bad block replacement strategies have the same result, but they are implemented differently. In the controller-initiated strategy for DSA disks, the hardware confirms and replaces a detected bad block. In the host-initiated strategy, the controller notifies the host software of a bad block. Then, the host software confirms and replaces the bad block.

The RX50, RX33, and RX23 devices have no means to replace bad blocks. If a hard error with a bad block is reported, the diskette must be replaced. First, try the diskette in another drive to be sure the problem is with the diskette and not the drive. If you can read the diskette, make a copy of it because the original diskette may be corrupt. File the original and use the copy. If the diskette is bad, recover as much of the data as you can and recreate the remainder of the diskette.

4.2 Error Correction Code Schemes

The disk controller detects and reports data errors when bad blocks cause erroneous data to be transferred from the drive to the controller. Error Correction Code (ECC) schemes are used to detect and correct any erroneous data that is found. These schemes apply statistical algorithms to dynamically correct erroneous data and to successfully complete requested I/O operations. If the data from a bad block is not correctable, the controller performs a series of rereads and other recovery techniques, and it attempts to obtain the data and transfer it without error.

Using ECC schemes, the hardware can correct an error when it is within a certain limit. If the error is equal to or below the limit, the controller can correct it and complete the I/O operation successfully. If the error is above the limit, the controller cannot correct the data, and, after repeated retries, the I/O operation fails and the failure is reported as a hard error to the host operating system, indicating that the block is bad and needs to be replaced.

Errors that are correctable but that are above a given threshold are also reported to the host operating system. These blocks are replaced to prevent future loss of data.

4.3 Bad Block Detection

The presence of a bad block on a disk may be indicated by one or more of the following occurances:

- Message displayed on terminal (DSA devices only)
- Entry in the errorlog file
- Output from the `radisk` utility or the `rzdisk` utility with the `-s` (scan) option specified
- Output from the `fsck` command

The logical block number (LBN) reported in an error message is relative to the start of the physical disk unit, and it is the LBN of the first block encountered. You should note the LBN associated with a bad block because you will need the LBN to replace the bad block either with the `radisk` utility or with the `rzdisk` utility.

Note

Defective hardware can cause good blocks to be reported as bad. The system may be replacing blocks that would not be considered bad if the hardware was functioning properly. Check your hardware first if an error report displays an unusual number of bad blocks.

4.4 The Errorlog File

When notified of a disk error, the host software enters an error record in the system errorlog file. To read logged error messages, you use the `uerf` command. For information on error logging and the errorlog file, see the *Guide to the Error Logger System* and `uerf`(8) in the *ULTRIX Reference Pages*.

The messages in the kernel errorlog buffer are logged to the errorlog file on disk, where they are accessed by the `uerf` command. The default errorlog file is `/usr/adm/syserr/syserr.`*hostname*.

The `uerf` command can access various disk errors from the errorlog file. To produce an error report containing all disk error messages from the errorlog file, run the `uerf` command with the `-D` option and no parameters.

Use the `uerf` command with the `-D` option and the disk parameter you want to access errors for the disk type that you specify.

Alternately, you can select errors by record type by specifying the `uerf` command with the `-r` option and the disk parameters that you want. For example, the following command selects disk errors:

```
# /etc/uerf -r 102 [RET]
```

The following command selects device controller errors:

```
# /etc/uerf -r 104 [RET]
```

The `-r` option does not produce error messages for the VAXstation 2000 or MicroVAX 2000 processors. To produce an error report with messages for the `sdc` driver, use the following command:

```
# /etc/uerf -c oper | grep sd? [RET]
```

Replace the question mark (?) in the preceding example with your drive unit number.

To diagnose a problem or to maintain disk integrity, study the types of errors logged. For example, a sudden incidence of hard errors reported at random places on a device may indicate an electrical or mechanical problem.

4.5 DSA Disk Error Reports

To obtain a report of DSA disk errors, use the following command:

```
# /etc/uerf -D ra [RET]
```

Examples of entries from the errorlog file for DSA devices are shown next.

The following error report indicates an error on a DSA device:

```
                                  uerf version 4.0-003 (113)


********************************* ENTRY     1. ********************************

----- EVENT INFORMATION -----

EVENT CLASS                                     ERROR EVENT
OS EVENT TYPE                      102.         DISK ERROR
SEQUENCE NUMBER                    884.
OPERATING SYSTEM                                ULTRIX 32
OCCURRED/LOGGED ON                              Tue Jan 23 11:09:36 1990 EST
OCCURRED ON SYSTEM                              ravine.zk3.d
SYSTEM ID                     x0A000005         MICROCODE REV = 5.
SYSTYPE REG.                  x02410201
                                                FIRMWARE REV = 65.
PROCESSOR TYPE                                  KA62A
PROCESSOR COUNT                      2.
PROCESSOR WHO LOGGED          x00000001

----- UNIT INFORMATION -----

UNIT CLASS                                      DSA DISK
UNIT TYPE                                       RA60
CONTROLLER NO.                       3.
UNIT NO.                           115.
ERROR SYNDROME                                  SDI ERROR

----- SDI INFO -----

PREVIOUS CYLINDER                    0.
CURRENT CYLINDER                  1599.
PREVIOUS HEAD                        4.
CURRENT HEAD                         4.
FRONT PANEL CODE                 x0096
```

The following error report indicates that a bad block was successfully replaced:

```
                              uerf version 4.0-006 (6)

********************************* ENTRY     1. ******************************

----- EVENT INFORMATION -----

EVENT CLASS                                 ERROR EVENT
OS EVENT TYPE                     102.      DISK ERROR
SEQUENCE NUMBER                   632.
OPERATING SYSTEM                            ULTRIX 32
OCCURRED/LOGGED ON                          Wed Jan  3 00:50:48 1990 EST
OCCURRED ON SYSTEM                          abyss
SYSTEM ID                    x06000068
SYSTYPE REG.                 x0000000B
                                            FIRMWARE REV = 0.
PROCESSOR TYPE                              KA88
PROCESSOR COUNT                     2.
PROCESSOR WHO LOGGED         x00000000

----- UNIT INFORMATION -----

UNIT CLASS                                  DSA DISK
UNIT TYPE                                   RA82
CONTROLLER NO.                      1.
UNIT NO.                           75.
ERROR SYNDROME                              BAD BLK REPL ATTMPT

----- BAD BLOCK INFO -----

BAD LBN                       1021511.
PREVIOUS RBN                        0.
NEW RBN                         17921.
BAD BLOCK REPL CAUSE             x01C8

----- SUPPORTING DATA -----

FORMAT                           x0009      BAD BLK REPL ATTMPT
FLAGS                            x0080
                                            OPERATION SUCCESSFUL
EVENT CODE                       x0014      BAD BLK REPLACEMENT
STATUS & EVENT CODE              x0014      BAD BLK SUCCESSFULLY REPLACED
BAD BLK REPL FLAG                x8000
                                            REPLACEMENT ATTEMPTED

----- ADDITIONAL INFO -----

UNIT ID                          x020B      RA82
                                            DEC STD 166 DISK DEVICE
COMMAND REF NUMBER           x1481003A
UNIT NUMBER                        75.
PACKET SEQUENCE NUM               739.
UNIQUE CONTR ID         x00000000FE01
CONTROLLER TYPE                  x0101      HSC50
                                            MASS STORAGE CONTROLLER
CONTROLLER SW/FW VER               39.
CONTROLLER HW VER                   0.
MULTI-UNIT                       x0062      PORT NUMBER: 98.
                                            REQUESTOR: 0.
UNIQUE UNIT NUMBER      x00000000D239
UNIT SOFTWARE VER                  33.
UNIT HARDWARE VER                   2.
VOLUME SERIAL NUMBER         x0001D56E
```

If you get an error report indicating that a bad block was not replaced, you should note the LBN of the bad block and use the `radisk` utility to replace the bad block.

4.5.1 Error Messages

The following sections describe the DSA error messages that can appear on the console terminal indicating a disk error.

Note

> The text of an ASCII error message, when reported in the brief or full output format, generates more than one entry in the errorlog file. These ASCII text messages may not print sequentially.

4.5.1.1 Bad Block Reported – When an error results from a bad block reported condition, a message appears on the error report. For example:

```
sd0: Bad Block Reported at LBN 79
      LBN 79 replaced
```

This message identifies the device and the location of the LBN and indicates that the block is bad and has been replaced.

4.5.1.2 Replace Failure – During a bad block replacement operation, data is read from the bad block and written to the replacement block. However, if a replacement operation fails, a message similar to the following appears on the error report:

```
       WARNING
REPLACE command failure at LBN 179368
on controller uq2 unit 1.  Notify field service
```

The message indicates that the controller failed to find the data at the new location on the disk. The unit is then software write-protected, which prevents further corruption of the disk. Data may be recovered from the disk after rebooting the system, but the data may be corrupt.

4.5.1.3 Forced Error Condition – If the data read from the bad block resulted in an uncorrectable ECC error, the replacement block is written with the forced error indicator. A forced error indicator warns that the data may be corrupted. A message is displayed when the hardware reads a block written with the forced indicator set. For example:

```
ra1a: hard error sn 12345
ra1a: Force Error Modifier set LBN 12345
```

The first line of this example appears at the console and the entire message is logged to the errorlog file.

This message indicates that the data in the block may be corrupted and should be restored. Restoring the data causes the block to be rewritten; this clears the forced error condition because the data is restored to a correct state.

When you discover a forced error indicator on a disk, you may not know what type of data is associated with the block generating the error. Before you restore a block reporting a forced error, you should perform the following steps to determine what kind of information the block contains and to recover data:

1. Use the `su` command to become superuser.
2. If the bad block is located on the system disk, use the `shutdown` command to bring the system to single-user mode.

   If the bad block is not located on the system disk, unmount the file system that contains the bad block.
3. Use the `icheck` command with the `-b` option to discover how the block is used.

   You need the sector number of the bad block from the errorlog message and the specification of the partition where the bad block occurred. Do not use the LBN. For example:

   ```
   # icheck -b 2300 /dev/rra0g  [RET]
   ```
4. If the `icheck` command produces an inode number, indicating the block is part of a file, specify the `ncheck` command with the `-i` option to determine the corresponding file. Use the inode number and the partition specification. For example:

   ```
   # ncheck -i 354 /dev/rra0g  [RET]
   ```

   Output from this command shows the file name or names associated with the bad block.
5. If the block with the error is a data file, copy the data to a new file and delete the old corrupted file. If the data is still corrupt, restore the file from a backup. Then use the `fsck` command to check and repair the file system. If the errors are still not cleared, run `fsck` again. You may need to run `fsck` several times.

   If the error is in the superblock, you should specify the `fsck` command with the `-b` option to use the backup superblock. For more information, refer to `fsck(8)` in the *ULTRIX Reference Pages*.

   If the error is in an inode or a cylinder group block, first try the `fsck` command. If `fsck` does not work, decide whether to restore the file system or replace the block.
6. If you decide to replace the block, use the appropriate disk maintenance utility.

4.6 SCSI Disk Error Reports

Each errorlog file entry provides the information needed to identify and locate a bad block on a SCSI disk, namely:

- The Additional Sense Code indicates the severity of the error condition. 0x17 specifies a recoverable error using data retries, 0x18 specifies a recoverable error using ECC schemes, and 0x11 specifies an unrecoverable error.
- The Device identifies the disk drive and partition (rz6g, for example) where the problem exists.
- The Sector Number specifies the offset from the beginning of the problem partition (20, for example).

- The Logical Block Address specifies the offset from the beginning of the disk (82864, for example).
- The LBN specifies the LBN of the first block encountered and is relative to the start of the physical disk unit.

To obtain a report of SCSI disk errors, use the following command:

```
# /etc/uerf -D rz RET
```

Examples of entries from the errorlog file for SCSI devices are shown next.

The following error report indicates a recovered disk error on a SCSI device:

```
                                          uerf version 4.0-006 (3)

********************************* ENTRY     1. ******************************

----- EVENT INFORMATION -----

EVENT CLASS                                ERROR EVENT
OS EVENT TYPE                      102.    DISK ERROR
SEQUENCE NUMBER                   5093.
OPERATING SYSTEM                           ULTRIX 32
OCCURRED/LOGGED ON                         Sun Feb 11 01:12:24 1990 EST
OCCURRED ON SYSTEM                         abyss
SYSTEM ID                     x06000068    HW REV: x1
                                           FW REV: x16
                                           CPU TYPE: R2000A/R3000
PROCESSOR TYPE                             KN01

----- UNIT INFORMATION -----

UNIT CLASS                                 SCSI DISK
UNIT TYPE                                  RZ23
CONTROLLER NO.                       0.
UNIT NO.                             4.

----- SCSI INFORMATION -----

REVISION                             2.
ERROR TYPE                       X0000     DEVICE ERROR
SUB_ERROR TYPE                       0.
SCSI ID                       x00220019    SCSI CMD PKT
                                           SCSI EXTENDED SENSE
                                           SCSI CTLR & DMA REGS
                                           SOFT ERROR
                                           DECSII
CMD BLK                          x000000000000001581B008
CMD                              x0008      READ

----- ADDITIONAL SCSI DATA -----

SCSI CTL SIGNAL 1                x0000
CTL STATUS REG                   x001F
DTAT REGISTER                    x0000
DMA INIT BYTE                    x0000
ID                               x8006
DMA ADR, LOW                     x0000
DMA ADR, HIGH                    x0001
DATA XFER STS                    x1104
CONN STATE                       x0400
SENSE KEY                        x0001    RECOVERED READ
```

```
ADD'L SENSE CODE              x0018
LOGICAL BLOCK NO.             7000.
SECTOR NO.                    7000.
XSENSE DATA
0000:    000100F0  0C581B00  00000000  00000018  *.....X.......*
0010:    02000000  00000000  00000000  00000000  *.............*
0020:    00000000  00000000  00000000
```

The following SCSI error report indicates that a bad block was successfully replaced:

```
                              uerf version 4.0-006 (6)

******************************** ENTRY     1. *****************************

----- EVENT INFORMATION -----

EVENT CLASS                                ERROR EVENT
OS EVENT TYPE                   102.       DISK ERROR
SEQUENCE NUMBER                   9.
OPERATING SYSTEM                           ULTRIX 32
OCCURRED/LOGGED ON                         Fri Jan 12 14:49:19 1990 EDT
OCCURRED ON SYSTEM                         nodename
SYSTEM ID                  x0A000005       MICROCODE REV = 5.
SYSTYPE REG.               x04010102
                                           FIRMWARE REV = 1.
PROCESSOR TYPE                             KA420

----- UNIT INFORMATION -----

UNIT CLASS                                 SCSI DISK
UNIT TYPE                                  RZ56
CONTROLLER NO.                    1.
UNIT NO.                          9.
BLOCK NUMBER                 986340.

----- SCSI INFORMATION -----

REVISION                          1.
ERROR TYPE                     x000D       DYNAMIC BAD BLK REPLACEMENT
SUB-ERROR TYPE            x00000002        REASSIGN BLOCK SUCCEEDED
SCSI ID                           1.
FLAGS                     x00120000
                                           SOFT ERROR
                                           NCR5380
```

If you get an error report indicating that a bad block was not replaced, you should note the LBN of the bad block and use the `rzdisk` utility to replace the bad block.

4.7 The Disk Maintenance Utilities

The disk maintenance utilities `radisk` and `rzdisk` are used on DSA and SCSI disks, respectively, and allow you to manually reassign bad blocks and to perform other system management duties such as scanning a disk for bad blocks. When the system reports that a bad block exists on your disk, you may have to reassign the block. To use the disk maintenance utilities, you must have superuser privileges and be in single-user mode. The disk must be unmounted.

4.7.1 The radisk Utility

The `radisk` utility allows you to manually replace, clear, and scan blocks on a DSA disk. The format for the `radisk` utility is:

radisk –option *LBN [length] special*

The `radisk` utility has five options:

- `-c` clear
- `-e` set exclusive access
- `-n` clear exclusive access
- `-r` replace
- `-s` scan

The *LBN* is the number of the logical block. The *length* is the number of blocks from the specified LBN that you want to scan or clear. Note that the `-r` option does not support the *length*. The *special* refers to the raw device special file (such as the `/dev/rra3c` device special file).

For the `-c` and `-r` options, the special file indicates an unmounted `c` partition of a character device special file. The `-s` option will accept any valid partition on the disk. The system produces an error message with both an LBN and a sector number.

The `-e` option sets the exclusive access attribute associated with the specified disk. This attribute is provided by multihost controllers to restrict access to a disk to one host.

The `-n` option clears the exclusive access attribute associated with the specified disk.

For more information on the `radisk` utility, refer to `radisk`(8) in the *ULTRIX Reference Pages.*

4.7.1.1 Clear Option Example – The `-c` option clears the forced error indicator in the specified area of a disk. The LBN can then be read without generating an error, even though the block may contain corrupted data.

The following command clears the forced error indicator from LBN 12334 to the last block of the partition:

```
# radisk -c 12334 -1 /dev/rra3c  RET
```

Use `-1` to refer to the end of a disk partition. In this example, `-1` represents the number of LBNs from LBN 12334 to the end of the disk.

Note

Clearing the block will remove the forced error indicator. However, the data in the block must still be considered corrupted. To maintain data integrity, check the data, and, if necessary, restore it from a backup. See the *Guide to System Backup and Restore* for information on restoring data.

4.7.1.2 Replace Option Example – The `-r` option forces the replacement of blocks on a disk. This option replaces a single block on a DSA disk. It enables you to save data in a block before it goes bad by replacing a block that records repeated transient errors.

For example, the following command replaces LBN 12345:

```
# radisk -r 12345 /dev/rra3c  RET
```

The `-r` option applies to host-initiated bad block replacement only; it does not apply to controller-initiated replacement.

Note

> Replacing blocks affects performance. As blocks are replaced, the speed of the system is affected. In addition, there are a limited number of replacement blocks available on a disk.

4.7.1.3 Scan Option Example – The `-s` option scans a disk for bad blocks. Use this option after reformatting or when the disk is new. The scan option reads the specified range of blocks and reports any errors. If a bad block is found, the block is replaced and the scan option restarts and rescans the specified range until no bad blocks are reported.

When a forced error condition is found, `radisk` reports the LBN and continues to scan. It does not restart and the forced error indicator is not cleared. Use the `-c` option to clear a forced error condition reported during a scan.

The following command scans the `g` partition starting at LBN 12345 for 2000 blocks:

```
# radisk -s 12345 2000 /dev/rra0g  RET
```

For the `-s` option, an LBN of 0 indicates the first LBN of the special file's partition.

4.7.2 The rzdisk Utility

When the system reports that a bad block exists on your disk, you may have to reassign the block. The following guidelines apply:

- If the errorlog file contains two or more daily entries for recoverable read data errors (additional sense code 0x17) for a particular block, you should reassign that block.
- If at any time the errorlog file contains entries for unrecoverable errors (additional sense code 0x11), you must reassign those blocks.
- If the errorlog file contains entries for ECC recoverable errors (additional sense code 0x18), you should not reassign the blocks. The device driver automatically reassigns blocks with ECC recoverable errors.

The `rzdisk` utility allows you to format a SCSI disk and to scan for and reassign bad blocks on a SCSI disk. You must have superuser privileges to run the `rzdisk` utility, and the system should be in single-user mode, except when formatting floppy diskettes.

For more information about the `rzdisk` utility, see `rzdisk(8)` in the *ULTRIX Reference Pages*.

The format of the `rzdisk` utility is:

rzdisk [-option] *[LBN] [length] [special]*

The `rzdisk` utility has four options:

`-f` format

`-h` help

`-r` reassign

`-s` scan

The *LBN* is a unique number (decimal notation) that represents the disk block as reported in the errorlog file.

The *length* is a decimal number that indicates how many 512-byte blocks to scan.

The *special* is a raw device pathname.

The `-h` option calls the HELP menu to the screen.

The `rzdisk` utility generates messages when the user is not privileged, when the LBN is not in the specified partition, and when the length exceeds the size of the partition.

4.7.2.1 Format Option Example – The `-f` option formats a SCSI hard disk. Note that since Digital ships the SCSI disk already formatted, you need to use this option with caution and only if the SCSI hard disk seems corrupted. Be aware that when you format a disk, all resident data is destroyed.

You can format a disk with the vendor (manufacturer) defect list or with the known (vendor and grown) defect list. The grown defect list contains any blocks that may have been reassigned during the life of the SCSI disk drive. When formatting a disk, you must specify the raw device pathname.

The following example reformats the disk on drive 3 using the known defect list:

```
# rzdisk -f known /dev/rrz3c  RET
```

4.7.2.2 Reassign Option Example – The `-r` option reassigns a bad block on the disk. The `-r` option is supported only by those SCSI disks that support the reassign block command. To reassign a bad block, you must specify the LBN and (raw) device `c` partition as arguments.

The `rzdisk` utility reads the specified LBN before reassigning the block. If it contains valid data, the block may not be bad, or the block may have been reassigned by the SCSI device driver. In this case, you should answer yes when `rzdisk` asks if you want to cancel the block reassignment. This prevents you from reassigning good blocks.

After reassigning the specified block, `rzdisk` displays a message to indicate whether the reassigned data is good or bad.

The following example reassigns block 222658 on the `c` partition of drive 3:

```
# rzdisk -r 222658 /dev/rrz3c  RET
```

The following command reassigns LBN 82864 on RZ55 unit 6:

```
# rzdisk -r 82864 /dev/rrz6c  RET
```

4.7.2.3 Scan Option Example – The `-s` option scans for bad blocks on a specified area of the disk.

When scanning a disk, you must specify the *LBN,* which is a unique number (decimal notation) that represents the disk block relative to the start of the partition, the *length,* and the raw device pathname of the partition to scan.

To start scanning from the first block of the specified partition, use the number 0 to represent the *LBN.* When the number 0 is specified, the scan starts at the first block of the specified partition.

The *length* is a decimal number that indicates how many 512-byte blocks to scan. To scan up to and including the last block of the specified partition, use the number `-1` to represent the *length.* By specifying the *length,* you define the scope of the scan within the identified partition.

The following example scans the first ten blocks of the entire disk (`c` partition) on drive 3:

```
# rzdisk -s /dev/rrz3c 0 10  RET
```

The following example scans the entire disk (`c` partition) on drive 3:

```
# rzdisk -s /dev/rrz3c 0 -1  RET
```

Device Mnemonics A

This appendix identifies and defines the mnemonics that are used to attach any hardware or software device to your system. The mnemonics are used by the `/dev/MAKEDEV` shell script to create the character or block special files that represent each of the devices. The mnemonics also appear in the system configuration file, as described in the *Guide to System Configuration File Maintenance.*

Table A–1 lists the mnemonics in nine categories: generic, systems, consoles, disks, tapes, terminals, modems, printers, and other. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The systems category lists the mnemonic for the DECstation 3100 setup. The consoles category lists mnemonics for the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices. The other category lists the mnemonic for the devices you can use with a DECstation 3100.

The description heading in Table A–1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to both the `MAKEDEV` script and the system configuration file, refer to the appropriate pages in Section 4 of the *ULTRIX Reference Pages.* If online reference pages are available, you can also use the `man` command. For instance, enter the following command at the system prompt to display the reference page for the Mass Storage Control Protocol (MSCP) disk controller driver:

```
% man ra
```

Where appropriate, the SYNTAX section of the reference page defines the device's syntax as it should appear in the `config` file. For additional software device mnemonics that `MAKEDEV` uses, refer to `MAKEDEV`(8) in the *ULTRIX Reference Pages.*

Note

Table A–1 uses the convention of an asterisk (*) beside a mnemonic and a question mark (?) beside a device name to mean a variable number. The value of the variable number is dependent on the particular device.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; for example, boot750
	mvax*	All MicroVAX setups; for example, mvax2000
	vaxstation*	A VAXstation 2000 setup; for example, vaxstation2000
	std	Standard devices with all console subsystems:
	drum	Kernel drum device
	errlog	Error log device
	kUmem	Kernel Unibus/Q–bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A tty device
	local	Customer-specific devices
Systems	DECstation	A DECstation 3100 setup
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives and RP?? devices
	ra*	UNIBUS/Q-bus/BI/HSC MSCP disk controller interface
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz*	SCSI disks (RZ22/RZ23/RZ24/RZ55/RZ56/RZ57/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI/HSC TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape
	tz*	SCSI tapes (TLZ04/TSZ05/TZ30/TZK50)
Terminals	cxa*	Q–bus cxa16
	cxb*	Q–bus cxb16
	cxy*	Q–bus cxt08
	dfa*	Q–bus DFA01 comm multiplexer
	dhq*	Q–bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer

Table A-1: (continued)

Category	Mnemonic	Description
	dhv*	Q–bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	dzq*	Q–bus DZQ11 comm multiplexer
	dzv*	Q–bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q–bus VCB02 (QDSS) graphics controller/console
	qv*	Q–bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
	lx	VAXstation 8000 color high-performance 3D graphics
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q–bus LP11 parallel line printer
Other	pm*	mono/color bitmap graphics/mouse/modem /printer/terminals for DECstation 3100

Index

U

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to Disk Maintenance
AA-ME93B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? ______________________________

____

____

What do you like best about this manual? ______________________________

____

____

What do you like least about this manual? ______________________________

____

____

Please list errors you have found in this manual:

Page Description

_____ ______________________________

_____ ______________________________

_____ ______________________________

_____ ______________________________

_____ ______________________________

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? __________

Name/Title ______________________ Dept. __________

Company ______________________ Date __________

Mailing Address ______________________________

__________ Email __________ Phone __________

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to Disk Maintenance
AA-ME93B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? ______________________________

____

____

What do you like best about this manual? ______________________________

____

____

What do you like least about this manual? ______________________________

____

____

Please list errors you have found in this manual:

Page	Description
______	______________________________
______	______________________________
______	______________________________
______	______________________________
______	______________________________

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? __________

Name/Title ______________________________ Dept. __________

Company ______________________________ Date __________

Mailing Address ______________________________

______________________________ Email ______________________________ Phone __________

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut Along Dotted Line